

IN THE CLAIMS:

1. A method of optimizing machine language code generated from source code having both single and double precision computation, said machine language code used to create an executable program, the optimizing machine language code comprising the steps of:

a) storing a count of times one or more precision computations are called in a double or single precision mode in a floating-point computation;

b) selecting a default precision mode to be either the double precision mode or the single precision mode and examining a frequency with which a target precision computation is called in a non-default precision mode, based on said stored count; and

c) generating and storing generalized code corresponding to a call in a non-default precision mode with respect to the target precision computation, based on said frequency with which a target precision computation is called.

2. The method of Claim 1, wherein step (a) further comprising a step of:
counting the number of times said precision computations are called in the non-default precision mode during runtime of said program.

3. The method of Claim 1, wherein step (b) further includes a step of comparing said count with a preset threshold and step (c) further includes a step of generating said generalized code when said count exceeds said threshold.

4. The method of Claim 1, wherein step (c) further comprising the steps of:
determining prior to generating said generalized code, whether a processing cost may be reduced if said target precision computation is operated in the non-default precision mode.

5. The method of Claim 1, wherein prior to generating and storing said generalized code, step (c) further comprises the steps of:

detecting any target precision computations in which the computation is executed in an excessive precision; and

replacing said detected target precision computations with single precision computations.

6. The method of Claim 1, wherein step (c) further comprises a step of performing code analysis with respect to the target precision computations to be compiled;

dividing the target precision computations into regions for executing single precision computations and regions for executing double precision computations, said regions being determined by results of the code analysis.

7. A method for generating a machine language code by controlling a computer to compile a program, the method comprising the steps of:

a) setting a computation precision for calling a target computation, said computation precision being based on a precision mode in a floating-point computation of a caller computation and generating and storing a machine language code;

b) determining the precision mode of said target computation and said caller computation, said precision mode being a double or single precision mode; and

c) generating an auxiliary code depending on the relation and adding the auxiliary code to the stored machine language code, if the precision mode in said caller computation is different from the precision mode in said target computation.

8. The method of Claim 7, wherein step (a) further comprises the steps of: detecting said target computation which is executed in an excessive precision; and replacing said precision mode of said target computation with a single precision mode.

9. The method of Claim 7, wherein step (a) further comprises the steps of:

performing code analysis of said target computation;
dividing said target computation into regions for executing single precision computations and regions for executing double precision computations in accordance with results of said analysis; and
setting said computation precision for each of said regions to generate the machine language code.

10. An apparatus for optimizing machine language code generated from source code having both single and double precision computations, said machine language code used to create an executable program, the apparatus comprising:

a compiler for compiling the source code according to a precision computation; and
a program execution unit for executing the source code in an interpreter and said program compiled by said compiler,

wherein said compiler

generates a first code of a target computation in a default precision mode selected from at least one of a double precision mode and a single precision mode in a floating-point computation, and

generates a second code of said target computation in a non-default precision mode selected from at least one of the double precision mode and the single precision mode if said target computation satisfies a predefined condition.

11. The apparatus of Claim 10, wherein if said target computation operating in said non-default precision mode calls another computation operating in said non-default precision mode, said program execution unit calls said second code generated in said compiler.

12. The apparatus of Claim 11, wherein if a target computation to be compiled may reduce processing costs by using said second code rather than said first code and if a frequency in

which said another computation is called from the target computation in said non-default precision mode is high, said compiler generates said second code.

13. The apparatus of Claim 10, wherein said compiler further comprises:

an excessive-precision optimization unit for detecting in a target computation a portion to be compiled in which a computation is executed in an excessive precision and replacing the computation of the portion with a single precision computation; and

a code generation unit for generating said first code and said second code to reflect a process result from said excessive-precision optimization unit.

14. The apparatus of Claim 10, wherein said compiler further comprises:

a precision region analysis unit for performing code analysis with respect to a target computation to be compiled and for dividing the target computation into regions for executing single precision computations and regions for executing double precision computations; and

a code generation unit for setting a computation precision for each of said regions to generate machine language code based on an analysis result from said precision region analysis unit.

15. A data processing apparatus for generating a machine language code by compiling a program, said apparatus comprising:

a code generation means for setting a computation precision, with respect to a target computation to be compiled, depending on a precision mode in a floating-point computation in a caller computation for calling said target computation, and generating a machine language code; and

auxiliary code addition means for generating an auxiliary code, based on a relation between the precision mode in said target computation and the precision mode in said caller computation, for matching said precision mode, and adding the auxiliary code to the machine language code generated in said code generation means.

16. The apparatus of Claim 15, wherein said auxiliary code addition means adds the code for matching the precision mode of the target computation with respect to said caller computation if the precision mode in said target computation is different from the precision mode in said caller computation.

17. The apparatus of Claim 15, further comprising excessive precision optimization means for detecting a portion in the target computation to be compiled in which the computation is executed in an excessive precision, and replacing the portion with a single precision computation,

wherein said code generation means generates machine language code to reflect a process result from said code addition means.

18. The apparatus of Claim 15, further comprising precision region analysis means for executing a code analysis with respect to the target computation to be compiled and dividing the target computation into regions for executing single precision computations and regions for executing double precision computations,

wherein said code generation means sets the computation precision for each of said regions to generate the machine language code, based on an analysis result from said precision region analysis means.

19. A program for generating a machine language code by controlling a computer to compile an executable program, said program causing said computer to perform the steps of:

a) storing information regarding whether a computation is called in a double precision mode or a single precision mode in a floating-point computation when said executable program is executed;

b) when a target computation is compiled, examining a frequency with which said target computation is called in a non-default precision mode, based on the stored information, if a

default precision mode is selected from at least one of the double precision mode and the single precision mode in the floating-point computation in said executable program; and

c) generating and storing a specialized machine language code corresponding to the call in said non-default precision mode with respect to said target computation based on obtained information on the frequency in which the target computation is called in said non-default precision mode.

20. The program of Claim 19, wherein step (c) further comprises a step of generating said specialized machine language code with respect to said target computation to be compiled if processing costs are reduced when said target computation is operated in said non-default precision mode.

21. The program of Claim 19, wherein step (c) further comprises the steps of:
detecting a portion in said target computation to be compiled, in which the computation is executed in an excessive precision; and
replacing the portion with a single precision computation.

22. The program of Claim 19, wherein step (c) further comprising the steps of:
performing code analysis with respect to said target computation to be compiled; and
dividing the target computation into regions for executing single precision computations and regions for executing double precision computations based on results of the code analysis.

23. A program for generating a machine language code by controlling a computer to compile an executable program, said program causing said computer to perform the steps of:

a) setting a computation precision, with respect to a target computation to be compiled, depending on a precision mode in a floating-point computation in a caller computation for calling said target computation and generating and storing a machine language code;

b) examining a relation between said target computation and said caller computation;
and

c) generating an auxiliary code depending on the relation and adding the auxiliary code to the stored machine language code.

24. The program of Claim 23, wherein step (c) further comprises the step of adding code for matching the precision mode of the target computation with respect to the caller computation if the precision mode in said target computation is different from the precision mode in said caller computation.

25. The program of Claim 23, further comprising the steps of:
detecting a portion to be compiled in said target computation, in which the computation is executed in an excessive precision; and
replacing the detected portion with a single precision computation.

26. The program of Claim 23, further comprising the steps of:
performing code analysis with respect to said target computation to be compiled;
dividing the target computation into regions for executing single precision computations and regions for executing double precision computations based on the results of the code analysis; and
setting the computation precision for each of said regions to generate the machine language code.

27. A computer program device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform computation steps for a program for generating a machine language code by controlling a computer to compile an executable program, said program causing said computer to perform the steps of:

a) storing information regarding whether a computation is called in a double precision mode or a single precision mode in a floating-point computation when said executable program is executed;

b) when a target computation is compiled, examining a frequency with which said target computation is called in a non-default precision mode, based on the stored information, if a default precision mode is selected from at least one of the double precision mode and the single precision mode in the floating-point computation in said executable program; and

c) generating and storing a specialized machine language code corresponding to the call in said non-default precision mode with respect to said target computation based on obtained information on the frequency in which the target computation is called in said non-default precision mode.

28. A computer program device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform computation steps for a program for generating a machine language code by controlling a computer to compile an executable program, said program causing said computer to perform the steps of:

a) setting a computation precision, with respect to a target computation to be compiled, depending on a precision mode in a floating-point computation in a caller computation for calling said target computation and generating and storing a machine language code;

b) examining a relation between said target computation and said caller computation; and

c) generating an auxiliary code depending on the relation and adding the auxiliary code to the stored machine language code.